

Sun ONE

Application Server

开发指南

师炜 译



We make the net work.

Sun™ ONE
Open Net Environment



机械工业出版社
China Machine Press

Sun ONE Application Server

开发指南

(美) Sun Microsystems, Inc 著
师 炜 译



机械工业出版社
China Machine Press

Sun ONE Application Server软件提供了一个最强劲的电子商务平台,用于向众多服务器、客户端和设备交付创新的先进应用服务。本书共分为6章,主要讲述如何在Sun ONE Application Server平台上创建和运行J2EE应用。主要内容包括:根据应用需求设计J2EE应用,开发J2EE应用,如何编写安全的J2EE应用,组装和部署J2EE应用,调试J2EE应用,开发Lifecycle监听器。本书除了讲述编程概念和开发任务外,还提供样例代码、实现技巧、参考资料和术语表。

本书的读者对象主要是在公司中开发、组装和部署J2EE应用的技术人员,对Sun ONE Application Server环境感兴趣的Java开发人员也可以阅读。

本书中文简体字版由Sun Microsystem, Inc授权机械工业出版社独家出版,未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Sun ONE Application Server开发指南 / 美国太阳微系统公司著;师炜译. -北京:机械工业出版社, 2003.6

书名原文: Developer's Guide: Sun ONE Application Server

ISBN 7-111-12132-5

I. S… II. ①太… ②师… III. ①网络服务器-程序设计 ②JAVA语言-程序设计
IV. ①TP368.5 ②TP312

中国版本图书馆CIP数据核字(2003)第035213号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:华章

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2003年6月第1版第1次印刷

787mm×1092mm 1/16·10.5印张

印数:0 001~3 000册

定价:42.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

关于 Sun ONE Application Server

Sun ONE Application Server（以前又称 iPlanet Application Server）软件提供一个最健壮的电子商务平台，用于向众多服务器、客户端和设备交付创新的先进应用服务。Sun ONE Application Server 建立在久经考验的世界级可扩展性、可靠性和高性能的基础上，是在用户需要稳夺市场先机，并对规模更大的企业内的信息系统和业务流程加以利用，以便成功开展全球电子商务活动的情况下应运而生的。Sun ONE Application Server 有 1000 多家客户，它可以让企业、服务提供商和 .com 企业抓住网络经济中的巨大发展机遇。

Sun ONE Application Server 的主要功能包括：

- 开发和交付基于 J2EE（Java 2 Platform, Enterprise Edition, Java 2 平台企业版）标准的应用。
- 通过高度可扩展、可靠和可用的体系结构来处理计划外的应用。
- 在一个平台上提供企业对企业、企业对消费者和企业对雇员的解决方案。
- 利用和集成企业内部和外部的信息资产。
- 让企业与供应商、客户和合作伙伴之间的业务流程实现自动化。

Sun ONE Application Server 为 J2EE 标准提供了强大的支持。几乎在每个开发和部署层面，Sun ONE Application Server 都具有这样的特色：即带有符合 J2EE 规范的界面和功能。另外，Sun ONE Application Server 平台实现了 J2EE 规范，同时具有重要业务性能，可以提供负载平衡、故障恢复和高可用性等服务，确保 J2EE 应用能够很容易地随着应用需求的增长而不断扩展。例如，Sun ONE Application Server 可以在 JSP 框架和 servlet 内提供负载平衡服务。同时，Sun ONE Application Server 完全通过了 J2EE CTS 认证，也就是说 Sun ONE Application Server 既是可移植的，也是可扩展的。

基于“开放工具”的策略，Sun ONE Application Server 平台的应用开发工作可以选择许多工具来完成，包括：

- Sun ONE Application Builder 提供了基于向导的开发，并与应用服务器紧密集成，大大提高了开发人员的生产效率。
- Forte for Java, Enterprise Edition 从一发布就将得到支持。这种高级集成开发环境（IDE）的特色是，可以开展企业级开发活动，带来团队开发能力，包括满足分布

式大规模应用调试需求。

- 第三方产品，比如 WebGain Visual Cafe、IBM VisualAge、Macromedia Dreamweaver 和 Inprise JBuilder，都可以插入到 Sun ONE Application Server 体系结构中，以保护开发投资。

Sun ONE Application Server 还提供了一套全面的基于 XML 的技术，促进 B2B 应用的集成。它还支持 Java Message Service，从而很容易地集成了异步消息发送方案。Java Message Service 为整个企业的关键业务数据和事件的异步交换提供了可靠、灵活的服务。

Sun ONE Application Server 解决方案能够在 J2EE 业务逻辑，以及包含在现有遗留企业系统中的业务逻辑中，实现业务流程自动化。iPlanet Process Manager 是一个基于 Web 的综合解决方案，用于设计、部署和管理 Sun ONE Application Server 上的业务流程。

为了与多个后台信息系统实现全面集成，Sun ONE Application Server 包括了 Unified Integration Framework 和 Enterprise Connectors，后者可以将 IBM CICS、BEA Tuxedo、SAP R/3 和 PeopleSoft 等 ERP 环境的应用和数据资产扩展为动态 Web 服务。

J2EE 采用的技术与标准

J2EE 定义了开发多层企业应用的标准。J2EE 通过将企业应用建立在标准化、模块化组件的基础上，通过为这些组件提供一组完整的服务，通过在不进行复杂编程的情况下自动处理许多应用操作的细节，从而大大简化了企业应用。

J2EE 利用了 J2SE 的许多特性，比如“一次编写，到处运行”的可移植性、用于数据库访问的 JDBC API、用于同现有企业资源交互的 CORBA 技术，以及即便在互联网应用中，也可以保护数据的安全模型。在这些技术的基础上，J2EE 增加了对 EJB 组件、Java Servlet API、JSP 和 XML 技术的全面支持。J2EE 标准包括了完整的规范和兼容性测试，确保在大量支持 J2EE 的现有企业系统之间，实现应用可移植性。

J2EE 的重要性

今天的企业通过快速开发和部署可以提供独特业务服务的应用，而赢得竞争优势。不管它们是用于提高雇员生产力的内部应用，还是针对专门客户或厂商服务的互联网应用，快速开发与部署都是成功的关键所在。

可移植性和可扩展性对于企业应用的长期生存也是很重要的。企业应用必须可以从小型工作原型和测试案例扩展到具有 24×7 服务能力的全企业服务，并能接受数十、数百甚至数千个客户同时访问。

然而，多层应用需要将一系列技术和资源、遗留数据和传统代码整合到一起。在今天的异构环境下，企业应用必须支持不同厂商提供的服务，而这些厂商拥有多样化的应用模型和其他标准。行业经验表明，集成这些资源需要花掉应用开发时间的 50% 以上。

作为建立在大量现有企业系统（比如数据库管理系统，事务监视器，命名和目录服务，等等）之上的一个标准，J2EE 打破了当前企业系统之间的内在障碍。统一的 J2EE 标准以一种统一的基于组件的应用模型，包容和接收了多层应用所需的现有资源。这使得用于满足企业战略需求的下一代组件、工具、系统和应用纷纷出现了。

J2EE 和 Sun ONE Application Server

Sun ONE Application Server（以前称为 iPlanet Application Server）提供了最健壮和最全面的 J2EE 规范的实现。J2EE 规范是服务器端 Java 应用的开放行业标准。对 J2EE 的支持确保可以在多个项目和多个平台上重复使用电子商务应用和组件。Sun ONE Application Server 完全支持 J2EE 规范，包括支持用于表示逻辑的 JSP、用于服务器端逻辑的 Java Servlet、用于核心业务逻辑的 EJB，以及用于访问关系数据库的 JDBC。由于 EJB 允许快速组装和部署应用，然后对其进行修改，并将其重新部署为基于互联网的新服务，因此对于网络经济中迅速变化的业务需求，EJB 是一种理想解决方案。Sun ONE Application Server 完全兼容 J2EE，为 Internet 应用和服务的开发、部署和重用提供了一个首选平台。

EJB 组件：Web 服务的基础

EJB 组件是为新的 Web 服务范例建立分布式业务功能的基础。全面的 EJB 规范为通过 J2EE 应用服务器，将强大的业务逻辑应用到多层企业应用环境提供了必要的条件。EJB 服务器端组件模型简化了中间件组件的开发，这些中间件组件是事务性、可扩展和可移植的。用 EJB 进行开发和通过应用服务器进行交付可以对事务处理、安全、数据库连接等中间件服务提供自动支持，从而降低了开发中间件的复杂性。

在当今瞬息万变的市场中，快速创建新的业务逻辑并利用现有 IT 资源交付这些逻辑，是交付服务的基本工作。JavaBean 组件体系结构表明，将一套完整的操作性能封装到易于配置和可以再用的客户端组件中是非常有用的。用 Java 语言编写的服务器端操作性能、能够访问现有企业系统的连接器，以及易于部署的模块化组件这三个概念的融合，产生了 EJB 组件标准，及其利用和整合全企业的中间件知识的性能。Sun 正在增强 EJB 组件的性能，让它们在一个灵活、动态的体系结构中发挥更大作用。这种体系结构能向众多客户和设备随处提供服务。

Sun 的随处交付 Web 服务策略的核心是 Sun 开放网络环境(Sun Open Net Environment,

Sun ONE)。开发人员关注的重点是一个新的体系结构。而这个体系结构的基础是一个智能 Web 服务构造，它用于提供可以在 Internet 上工作的应用。Sun 将智能服务定义为能够确定用户，并能考查他在哪里、他与谁在一起以及他正在做什么这样的 Web 服务。智能 Web 服务考虑了时间、地点和其他环境元素。

Web 服务代表一个可以通过 Web 进行访问的业务、应用或系统功能单元。Web 服务适用于任何类型的 Web 环境，可以是 Internet、内部网或者外部网。它专注于进行企业对企业、企业对企业、部门对部门或点对点通信。Web 服务消费者可以通过一台桌面电脑或者无线浏览器来访问服务的用户，是一个应用程序，也可以是另一种 Web 服务。

智能 Web 服务的功能：

- 可通过 Web 访问。
- 公开 XML 接口。
- 已注册并且能够通过 Web 服务注册进行查找。
- 用 XML 消息在标准 Web 协议上进行通信。
- 支持系统间的松散耦合连接。

Web 服务使用 XML 接口和 XML 消息，在标准 Web 协议上进行通信，因此所有 Web 服务环境都是可以互操作的。

使用 EJB 组件，可以显著简化开发、部署和维护智能 Web 服务。它使得开发人员可以将注意力集中到业务逻辑编程的细节（这是开发人员的核心技能），同时依靠各种后台服务来提供基础设施，依靠客户端应用来提供强大的用户交互性能。完成开发后，就可以将业务逻辑部署到服务器上。这些服务器适合某一机构的现有需求，或该机构所服务的市场。

可以从以下网址免费下载 Sun ONE Application Server 7:

<http://gceclub.sun.com.cn>

亲身体验 Sun ONE Application Server

我使用新的 Sun One Application Server 7 标准测试版已有大约一个月了，感觉非常好。Sun 对以前的 Application Server 的改写非常成功。我发现其安装、配置、开发和维护都不费神。对于该应用服务器的每个部分，都提供了相应文档。其错误日志非常详细。另外，它还易于同其他 Sun One 技术（例如 Directory Server、Identity Server，等等）集成，因此，对于解决一个小项目变成大项目时带来的越来越多的苦恼，它是一种理想选择。下面，我详细介绍使用这个应用服务器的亲身体验。

服务器的安装毫不费力。我从 Sun 网站上下载了 Sparc 测试版，解压到一个临时目录。然后继续运行安装命令，经过很少的几幅对话框屏幕后就安装完成了。标准安装包括 Java 1.4.0（当然你也可以使用你自己的）和用于示例应用的 PointBase 数据库（它的数据限制为 5MB，如果没有必要的话，你可以不安装它）。在安装中还包括一个完整的示例目录。通过逐步回答对话框的问题，用了不到 5 分钟的时间，就将应用服务器就安装到了我的 Sun Ultra 2 机器上。

安装完应用服务器后，我开始配置它。有 3 种方式来管理和配置应用服务器：Web、命令行和 ant。确实，通过 ant 你可以用应用服务器来完成任何任务。为此，Sun 在应用服务器的安装中包括了一个 ant 的扩展版本。不过就我个人来说，喜欢使用 Apache 的晚间构建（nightly build），因此我只拷贝了两个 Sun Ant 文件到我的 ant 库目录，然后我就离开了。由于你可以用 ant 完成几乎所有事情，于是我创建了一个 ant 任务来配置应用服务器。事实证明，这非常有用，因为我的公司正在进行新硬件的评估，大家都很忙。使用 ant，我可以在 15 分钟内安装好 Application Server，并让它开始运行。

管理和配置该服务器的第二种方式是使用命令行。Sun 提供了一个名为 asadmin 的命令程序。通过这个程序，你可以使用该应用服务器上的任何功能。命令语法经过精心的文档化整理，其多模式特性允许你仅输入一次命令参数，而不管你需要运行多少条命令。如果你试图连接到远程实例，这真的很节省时间。

管理和配置该应用服务器的第三种方式是通过 Web 前端工具。Web 前端工具给我的印象十分深刻。像使用大多数应用服务器一样，你可以部署、解除部署、检查状态、添加和删除资源，等等。给我印象最深刻的是，由于 Web 前端工具具有直观性，所以我们的操作

管理员可以非常方便地对服务器进行维护。另外,我们的管理员喜欢用应用服务器的 SNMP 能力来监控它。他们目前正在将其捆绑到数据中心监控套件中。

在文档中包含的是经过验证的快速启动指南。第一步是配置 DataSource 缓冲池。Sun 提供了一个好的数据源预配置选择。我目前使用 Oracle 9i, 只需要加入用户名、密码等数据。第二步是设置 JDBC 数据源, 使之连接到缓冲池。最后一步配置工作是制作持久化管理器, 供实体 bean 使用。这一步很简单, 完成得很顺利。其好的特性是具有使用第三方持久化管理器的和运行多个持久化管理器的性能。这是需要你完成的惟一配置, 它需要 10 分钟左右。后来, 我按照文档很容易地配置了一个 RDBMS 安全域。让我印象特别深刻的是, Liberty Alliance 的单点登录功能和它与 LDAP 的集成。我还须对此予以评估, 看来它非常有前途。

安装完 Application Server, 配置好数据库连接后, 我开始编码。我的当前应用使用了 CMP/CMR EJB、Session EJB、JSP、Servlet 和 Taglib。我还使用了 Apache Struts (不能等 Java Server Faces) 来处理 Web 层。对于集成开发环境, 我没有用。我使用 ksh、vi 和 ant 来完成所有事情。至此我已经部署了 3 个本地实体 bean、两个会话 bean、两个 taglib、5 个 Struts 活动类、两个过滤器, 以及 20 个服务器页面。就像任使用何其他应用服务器一样, 我在这个过程中学到了不少东西。

就像使用任何好架构一样, 我从域模型和实体 bean 开始。Application Server 需要一个服务器专有的 XML 文件, 详细记录要使用的持久化管理器。另外, 还需要一个映射 XML 文件, 用于将 bean 中的实例域映射到数据库表的列。需要做这些额外工作的原因在于, Application Server 不能自动创建表。我刚开始认为这很特别, 是因为大多数应用服务器都能够创建表。经过仔细考虑后, 我很少使用这个功能。我经常首先创建表或者使用现有的表或者数据。不过我还是希望 Application Server 能有制表能力。除了需要这两个额外的 XML 文件外, 你还必须将数据库模式捕获到一个文件中, 并将其加进 EJB 中。将数据库模式读入文件, 允许在没有数据库连接的情况下, 建立和测试 Sun 集成开发环境。如果你需要的话, 这是一个好的特性, 如果你不需要的話, 这就是一个不必要的步骤。一旦创建了两个必需的 XML 文件, 并捕获了数据库模式, 部署和使用 CMP EJB 就没有任何问题了。我还创建了一些 CMR EJB, 并且在对 XML 文件作了更改后, 它们的工作状态良好。我不在意需要额外的 XML 文件, 但确实不喜欢在没有集成开发环境的情况下自己创建它们。我目前正在创建一个用于生成 XML 文件的 ant 任务。开发会话 bean 的过程中没有遇到什么困难。Servlet、JSP 和 taglib 库也正如所希望的那样运行。我大量使用过 Jakarta Struts 1.1, 没发现什么不同的操作性能。

评估了 SunOne Application Server 7 以后，我打算在以后的开发工作中使用它。它对一个 CPU 免费发放许可，并且可以在多个平台（Solaris、AIX、HP-UX、Linux、Windows 等）上使用，这使它成了一个非常有吸引力的产品。另外，企业版的故障容忍性能好像也领先于其他同类产品。我建议你在未来的项目中试用这个 Application Server。Solaris 和 Windows 版本的暂定发布日期为 10 月 28 日。预计其他操作系统版本要晚 3 个月发布。通过访问网址 [http://see.sun.com/cgi-bin/gx.cgi/mcp?q=ST1D00uTE\\$zZeuh](http://see.sun.com/cgi-bin/gx.cgi/mcp?q=ST1D00uTE$zZeuh)，就可以在它发布时收到一封电子邮件；还可以通过以下网址获得更多信息：

http://www.sun.com/software/products/appsrvr/home_appsrvr.html

<http://sunonedev.sun.com/>

——转摘自 http://www.theserverside.com/reviews/thread.jsp?thread_id=16108,

作者：Matt Bauer

前 言

本书描述如何创建和运行 J2EE (Java 2 Platform, Enterprise Edition) 应用, 这些应用遵循 Servlet、Enterprise JavaBean (EJB 组件) 和 Sun Open Net Environment (Sun ONE) Application Server 7 上的 JavaServer Page (JSP) 新的开放的 Java 标准模型。除了描述编程概念和任务外, 本书还提供样例代码、实现技巧、参考资料和术语表。

前言包括有关以下主题的信息:

- 谁应用使用本书。
- 使用文档。
- 本书的组织结构。
- 相关信息。
- 本书约定。
- 产品支持。

谁应该使用本书

本书的读者是在公司中开发、组装和部署 J2EE 应用的技术人员。

本书假设你熟悉以下主题:

- J2EE 规范。
- HTML。
- Java 编程。
- 在 servlet、JSP、EJB 和 JDBC 规范中定义的 Java API。
- 结构化数据库查询语言, 如 SQL。
- 关系数据库概念。
- 软件开发过程, 包括调试和源代码控制。

使用文档

Sun ONE Application Server 手册可以在线获得, 文件格式为 PDF 和 HTML, 网址为:

<http://docs.sun.com/>

表 0-1 列出 Sun ONE Application Server 手册中描述的任务和概念。左边一栏列出了任务和概念，右边一栏列出了相应的手册。

表 0-1 Sun ONE Application Server 文档一览表

有关信息	参 见
关于软件和文档的最新信息	<i>Release Notes</i>
支持的平台和环境	<i>Platform Summary</i>
对应用服务器的介绍，包括新特性、评价安装信息和总体概览	<i>Getting Started Guide</i>
安装 Sun ONE Application Server 及其各种组件（样例应用、Administration 界面、Sun ONE Message Queue）	<i>Installation Guide</i>
创建和实现遵循 Sun ONE Application Server 7 上开放 Java 标准模型的 J2EE 应用。包括关于应用设计、开发者工具、安全、组装、部署、调试和创建生命周期模块的一般信息	<i>Developer's Guide</i>
创建和执行遵循 Sun ONE Application Server 7 上关于 Web 应用的开放 Java 标准模型的 J2EE 应用。讨论 Web 应用编程概念和任务，并提供样例代码、实现技巧和参考资料	<i>Developer's Guide to Web Applications</i>
创建和实现遵循 Sun ONE Application Server 7 上关于企业 bean 的开放 Java 标准模型的 J2EE 应用。讨论 EJB 编程概念和任务，并提供样例代码、实现技巧和参考资料	<i>Developer's Guide to Enterprise JavaBeans Technology</i>
创建访问 Sun ONE Application Server 7 上 J2EE 应用的客户端	<i>Developer's Guide to Clients</i>
创建 Web 服务	<i>Developer's Guide to Web Services</i>
J2EE 特性，比如 JDBC、JNDI、JTS、JMS、JavaMail、资源和连接器	<i>Developer's Guide to J2EE Features and Services</i>
创建自定义 NSAPI 插件	<i>Developer's Guide to NSAPI</i>

有 关 信 息	参 见
执行以下管理任务： ● 使用 Administration 界面和命令行界面 ● 配置服务器预置参数 ● 使用管理域 ● 使用服务器实例 ● 监视和记录服务器活动 ● 配置 Web 服务器插件 ● 配置 Java Messaging Service ● 使用 J2EE 特性 ● 配置对基于 CORBA 客户端的支持 ● 配置数据库连接 ● 配置事务管理 ● 配置 Web 容器 ● 部署应用 ● 管理虚拟服务器	<i>Administrator's Guide</i>
编辑服务器配置文件	<i>Administrator's Configuration File Reference</i>
配置和管理 Sun ONE Application Server 7 操作环境的安全性。包括关于一般安全、验证和 SSL/TLS 加密的信息。还解决了 HTTP 基于服务器的安全问题	<i>Administrator's Guide to Security</i>
配置和管理 Sun ONE Application Server 7 关于 J2EE CA 连接器的服务提供者实现。包括关于 Administration Tool、DTD 的信息，并提供样例 XML 文件	<i>J2EE CA Service Provider Implementation</i> <i>Administrator's Guide</i>
将应用从 Netscape Application 2.1 迁移到新的 Sun ONE Application Server 7 编程模型，包括 Sun ONE Application Server 提供的 Online Bank 应用的样例迁移	<i>Migration Guide</i>
使用 Sun ONE Message Queue	Sun ONE Message Queue 文档位于： http://docs.sun.com/?p=/coll/S1_MessageQueue_30

本书的组织结构

本书为程序设计提供了 Sun ONE Application Server 环境的概览，包含以下主题：

- 第 1 章，“设计应用”。

总结了 Sun ONE Application Server 应用设计过程，并提供有效的开发指南。

- 第 2 章，“开发 J2EE 应用”。

描述如何设置开发环境并为创建应用组件提供基本步骤。

- 第 3 章，“编写安全的 J2EE 应用”。

描述如何编写安全的 J2EE 应用，这种应用中包含执行用户身份认证和对 servlet 和 EJB 业务逻辑的访问授权的组件。

- 第 4 章，“组装和部署 J2EE 应用”。

描述 Sun ONE Application Server 模块的内容，以及如何将这些模块单独或一起组装在一个应用中。

- 第 5 章，“调试 J2EE 应用”。

给出在 Sun ONE Application Server 7 中调试应用的指南。

- 第 6 章，“开发 Lifecycle 监听器”。

描述如何创建和使用生命周期模块，该模块在服务器启动时自动初始化并通报服务器的关闭。

最后是术语表。

相关信息

你可以在 `install_dir/docs/index.htm` 处找到官方规范的 URL 目录。另外，我们还推荐以下资源：

一般的 J2EE 信息：

由 Prentice Hall 出版，Deepak Alur, John Crupi 和 Dan Malks 编写的 *Core J2EE Patterns: Best Practices and Design Strategies*。

由 O'Reilly 出版, Scott Oaks 编写的 *Java Security*。

利用 Servlet 和 JSP 编程:

由 O'Reilly 出版, Jason Hunter 编写的 *Java Servlet Programming*。

由 O'Reilly 出版, Scott Oaks 和 Henry Wong 编写的 *Java Threads, 2nd Edition*。

利用 EJB 组件编程:

由 O'Reilly 出版, Richard Monson-Haefel 编写的 *Enterprise JavaBeans*。

利用 JDBC 组件编程:

由 O'Reilly 出版, George Reesel 编写的 *Database Programming with JDBC and Java*。

由 Graham Hamilton, Rick Cattell 和 Maydene Fisher 编写的 *JDBC Database Access With Java: A Tutorial and Annotated Reference (Java Series)*。

本书约定

本节描述本书通篇使用的几种约定:

- 一般约定。
- 有关目录的约定。

一般约定

本书使用以下一般约定:

- **文件和目录路径**以 UNIX 格式给出(以前向斜杠分隔目录名称)。对于 Windows 版本, 目录路径除了以反斜杠分隔目录之外其余是相同的。
- **URL**以 `http://server.domain/path/file.html` 这样的格式给出。

在这些 URL 中, *server* 是应用运行所在的服务器名; *domain* 是你的 Internet 域名; *path* 是服务器的目录结构; *file* 是有含义文件名。URL 中的斜体部分是占位符。

● 字体约定包括:

- 等宽字体 (monospace) 用于范例代码和代码清单、API 和语言元素(比如函数名和类名)、文件名、路径名、目录名和 HTML 标记。
- 斜体 (*Italic*) 用于代码变量、占位符、书名、手册名。
- 粗体 (**Bold**) 用于重点内容。

- **安装根目录**在本书中对于多数平台都用 *install_dir* 指示。“有关目录的约定”中列举了例外情况。

默认情况下，多数平台上 *install_dir* 的位置是：

- Solaris 8 不基于软件包的评估安装：
user's home directory/sun/appserver7
- Solaris 非捆绑的、非评估安装：
/opt/SUNWappserver7
- Windows 全部安装：
C:\Sun\AppServer7

对于上面列出的平台，*default_config_dir* 和 *install_config_dir* 都等同于 *install_dir*。有关例外情况和附加信息，请参见“有关目录的约定”。

- **实例根目录**在本书中用 *instance_dir* 来表示，它是以下目录的缩写：
default_config_dir/domains/domain/instance
- **本书中专用于 UNIX 的描述**也适用于 Linux 操作系统，特别提到 Linux 的地方除外。

注意 Forte for Java 4.0 已改名为 Sun ONE Studio 4。

有关目录的约定

默认情况下，当使用 Solaris 8 和 9 基于软件包的安装及 Solaris 9 捆绑销售的安装时，应用服务器文件是跨几个根目录分布的。本节将描述这些目录。

- **对于 Solaris 9 捆绑销售的安装**，本书使用以下文档约定与所提供的各种默认安装目录对应：
 - *install_dir* 是指 */usr/appserver/*，其中包含安装图像的静态部分。组成应用服务器的所有实用工具、可执行程序 and 库都在这个位置。
 - *default_config_dir* 是指 */var/appserver/domains*，这是所创建的任何域的默认位置。
 - *install_config_dir* 是指 */etc/appserver/config*，其中包含安装范围的配置信息，比如许可证和为该安装配置的管理域的主列表。
- **对于 Solaris 8 和 9 基于软件包的、非评估的、非捆绑的安装**，本书使用以下文档约定与所提供的各种默认安装目录对应：
 - *install_dir* 是指 */opt/SUNWappserver7*，其中包含安装图像的静态部分。组成应用服务器的所有实用工具、可执行程序 and 库都在这个位置。
 - *default_config_dir* 是指 */var/opt/SUNWappserver7/domains*，这是所创建

的任何域的默认位置。

- *install_config_dir* 是指 `/etc/opt/SUNWappserver7/config`, 其中包含安装范围的配置信息, 比如许可证和为该安装配置的管理域的主列表。

产品支持

可以从以下网址免费下载 Sun ONE Application Server 7:

<http://gceclub.sun.com.cn>

如果你的系统有问题, 那么请用以下任一种方法联系客户支持:

- 位于 <http://www.sun.com/supporttraining/> 的在线支持 Web 站点。
- 你的维护合同上相关的电话号码。

请在联系客户支持之前提供以下信息, 这样有助于确保我们的支持人员可以最好地帮助你解决问题:

- 问题的描述, 包括问题发生的情形以及对操作的影响。
- 机器类型、操作系统版本和产品版本, 包括可能引起问题的任何补丁和其他软件。
- 导致重复产生问题的详细步骤。
- 任何错误日志或内核溢出。